(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－187558

(43)公開日　平成10年(1998)７月21日

(51)Int.Cl.[6]		識別記号	FI		
Ｇ０６Ｆ	13/00	３５１	Ｇ０６Ｆ	13/00	３５１Ｃ
	9/46	３６０		9/46	３６０Ｆ
	15/16	４６０		15/16	４６０Ｔ

審査請求　未請求　請求項の数７　ＯＬ　（全 13 頁）

(21)出願番号　特願平8－339717

(22)出願日　平成８年(1996)12月19日

(71)出願人　000001443
カシオ計算機株式会社
東京都渋谷区本町１丁目６番２号

(72)発明者　矢代　義徳
東京都羽村市栄町３丁目２番１号　カシオ計算機株式会社羽村技術センター内

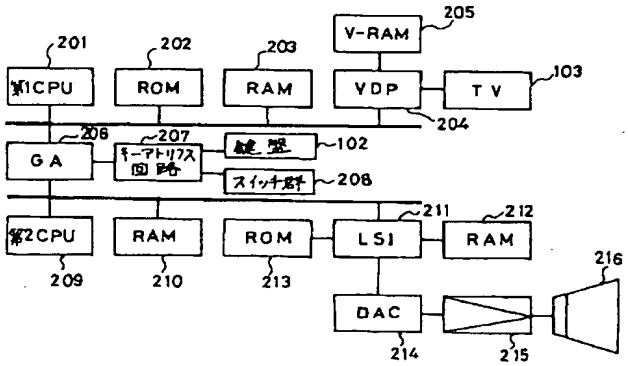
(74)代理人　弁理士　阪本　紀康

(54)【発明の名称】　データ転送方法、及びデータ処理装置

(57)【要約】

【課題】　マルチプロセッサ方式を採用した装置等において、データ転送に要する時間を短縮して処理の効率を向上させる。

【解決手段】　第１ＣＰＵ２０１は、ＲＯＭ２０２内のプログラムを実行することにより、スイッチ処理、鍵盤処理といった主要なサブルーチン処理、それらのサブルーチン処理を実行することで生成した発音制御データを第２ＣＰＵ２０９に転送する処理を行う。そのデータを転送する処理では、ＲＯＭ２０２に格納された楽音制御データ処理時間データ群や主要ルーチン実行時間データ群等から、再び転送処理を実行するまでに処理が完了しないだけの発音制御データが第２ＣＰＵ２０９に転送されていると予測した場合、第２ＣＰＵ２０９に転送すべきデータの転送を行わずに他のサブルーチン処理に移行する。



第１の実施の形態が適用されたシステムの構成図

1

【特許請求の範囲】

【請求項１】　第１の処理手段が第２の処理手段に、該第２の処理手段に処理させるデータを転送する方法であって、

前記第２の処理手段が処理に必要とする処理時間をデータの種類別に予め用意し、

前記第２の処理手段に転送すべきデータが生じた場合に、前記処理時間から前記第２の処理手段が転送済みのデータに対する処理を完了するまでの処理残り時間を予測し、

該予測した処理残り時間に基づいて、前記転送すべきデータを前記第１の処理手段から前記第２の処理手段に転送させる、

ことを特徴とするデータ転送方法。

【請求項２】　第１の処理手段が第２の処理手段に、該第２の処理手段に処理させるデータを転送する方法であって、

前記第２の処理手段が処理に必要とする処理時間をデータの種類別に予め用意し、

前記第１の処理手段がルーチン処理の実行に必要とする実行時間をルーチン処理の種類別に用意し、

前記第１の処理手段が前記第２の処理手段にデータ転送を行う転送処理を前記ルーチン処理として実行したとき、前記第２の処理手段に転送すべきデータが生じていた場合に、前記処理時間から前記第２の処理手段が転送済みのデータの処理を完了するまでの処理残り時間、及び、前記実行時間を参照して、前記転送処理を再び実行するまでの実行間隔をそれぞれ予測し、

前記処理残り時間、及び実行間隔の各予測結果に基づいて、前記転送すべきデータを前記第１の処理手段から前記第２の処理手段に転送させる、

ことを特徴とするデータ転送方法。

【請求項３】　前記実行時間は、前記第１の処理手段が各ルーチン処理の実行に要した実際の実行時間を計測して取得する、

ことを特徴とする請求項２記載のデータ転送方法。

【請求項４】　所定の第１の処理、及び該第１の処理の実行により得た転送すべきデータを転送する転送処理をルーチン処理として少なくとも実行する第１の処理手段と、該第１の処理手段から転送されたデータを用いて処理を実行する第２の処理手段とを備えたデータ処理装置において、

前記第１の処理手段が前記転送処理を実行したとき、前記第２の処理手段が転送済みのデータの処理を完了するのに必要とする処理残り時間を予測する第１の予測手段と、

前記第１の予測手段が予測した処理残り時間に基づいて、前記第１の処理手段による前記第２の処理手段へのデータ転送を制御する制御手段と、

を具備したことを特徴とするデータ処理装置。

2

【請求項５】　前記第１の処理手段が前記転送処理を実行してから次にそれを実行するまでの実行間隔を予測する第２の予測手段を更に具備し、

前記制御手段は、前記第１及び第２の予測手段の各予測結果に基づいて、前記第１の処理手段による前記第２の処理手段へのデータ転送を制御する、

ことを特徴とする請求項４記載のデータ処理装置。

【請求項６】　前記第２の予測手段は、前記第１の処理手段が実行した前記転送処理の実際の実行間隔を計測して取得し、該取得した過去の実行間隔に基づいて、前記実行間隔を予測する、

ことを特徴とする請求項５記載のデータ処理装置。

【請求項７】　前記制御手段は、前記第１及び第２の予測手段の少なくとも一つの予測結果に基づいて転送させるべきデータの種類を選択し、該選択した種類のデータを前記転送すべきデータのなかから抽出して前記第１の処理手段に転送させる、

ことを特徴とする請求項４、５、または６記載のデータ処理装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、転送先の負荷の重さを予測してデータを転送する技術に関する。

【０００２】

【従来の技術および発明が解決しようとする課題】例えば電子楽器では、近年多機能化が進み、数多くの機能が搭載される傾向にある。楽音を発音するという本来の機能の他に、音符や絵を表示させる画像表示機能等が搭載されることも多くなっている。

【０００３】搭載される機能が多くなるほど、全体の処理の負荷は大きくなる。搭載される機能を実現させるための処理を１つのＣＰＵに集中させると、一通りの処理を実行するのに要する時間が長くなり、動作が遅くなるという不具合が発生する。そのため、電子楽器においても、搭載された機能を実現するために、マルチプロセッサ方式が採用されることが多くなっている。

【０００４】そのマルチプロセッサ方式では、あるＣＰＵ（以降、他との区別のために第１ＣＰＵと呼ぶ）が他のＣＰＵ（以降、他との区別のために第２ＣＰＵと呼ぶ）にデータを転送し、処理を依頼する形になる。従来のデータ転送方法では、第１ＣＰＵから第２ＣＰＵへのデータ転送を、第２ＣＰＵ側に用意されているデータ格納領域（バッファ）の空容量を確認して行われていた。具体的には、例えば第２ＣＰＵにデータ転送を行う場合、第１ＣＰＵは第２ＣＰＵにデータ格納領域の空容量を問い合わせ、転送すべきデータを格納できるだけの容量を第２ＣＰＵが用意したのを確認した後、データ転送を行っていた。

【０００５】しかし、上記のデータ転送方法では、空容量の有無を確認する期間や第２ＣＰＵが空容量の確保を

行っている期間、第１ＣＰＵは実質的に待ち状態となる。そのため、データ転送に長い時間がかかり、それによってスループットが低下することから、処理を効率的に行えないという問題点があった。

【０００６】本発明の課題は、データ転送に要する時間を短縮して処理の効率を向上させることにある。

【０００７】

【課題を解決するための手段】本発明の第１の態様のデータ転送方法は、第１の処理手段が第２の処理手段に、該第２の処理手段に処理させるデータを転送するためのものであり、第２の処理手段が処理に必要とする処理時間をデータの種類別に予め用意し、第２の処理手段に転送すべきデータが生じた場合に、処理時間から第２の処理手段が転送済みのデータに対する処理を完了するまでの処理残り時間を予測し、該予測した処理残り時間に基づいて、転送すべきデータを第１の処理手段から第２の処理手段に転送させる。

【０００８】本発明の第２の態様のデータ転送方法は、第１の処理手段が第２の処理手段に、該第２の処理手段に処理させるデータを転送するためのものであり、第２の処理手段が処理に必要とする処理時間をデータの種類別に予め用意し、第１の処理手段がルーチン処理の実行に必要とする実行時間をルーチン処理の種類別に用意し、第１の処理手段が第２の処理手段にデータ転送を行う転送処理をルーチン処理として実行したとき、第２の処理手段に転送すべきデータが生じていた場合に、処理時間から第２の処理手段が転送済みのデータの処理を完了するまでの処理残り時間、及び、実行時間を参照して、転送処理を再び実行するまでの実行間隔をそれぞれ予測し、処理残り時間、及び実行間隔の各予測結果に基づいて、転送すべきデータを第１の処理手段から第２の処理手段に転送させる。

【０００９】なお、上記実行時間は、第１の処理手段が各ルーチン処理の実行に要した実際の実行時間を計測して取得する、ことが望ましい。本発明の第１の態様のデータ処理装置は、所定の第１の処理、及び該第１の処理の実行により得た転送すべきデータを転送する転送処理をルーチン処理として少なくとも実行する第１の処理手段と、該第１の処理手段から転送されたデータを用いて処理を実行する第２の処理手段とを備えていることを前提として、第１の処理手段が転送処理を実行したとき、第２の処理手段が転送済みのデータに対する処理に必要とする処理残り時間を予測する第１の予測手段と、第１の予測手段が予測した処理残り時間に基づいて、第１の処理手段による第２の処理手段へのデータ転送を制御する制御手段と、を具備する。

【００１０】本発明の第２の態様のデータ処理装置は、上記第１の態様の構成に加えて、第１の処理手段が転送処理を実行してから次にそれを実行するまでの実行間隔を予測する第２の予測手段を更に具備し、制御手段は、

第１及び第２の予測手段の各予測結果に基づいて、第１の処理手段による第２の処理手段へのデータ転送を制御する。

【００１１】なお、上記の構成において、第２の予測手段は、第１の処理手段が実行した転送処理の実際の実行間隔を計測して取得し、該取得した過去の実行間隔に基づいて、実行間隔を予測する、ことが望ましい。また、制御手段は、第１及び第２の予測手段の少なくとも一つの予測結果に基づいて転送させるべきデータの種類を選択し、該選択した種類のデータを転送すべきデータのなかから抽出して第１の処理手段に転送させる、ことが望ましい。

【００１２】本発明では、転送元がデータの転送先にデータを転送する場合、転送先の処理能力、転送済みのデータ量、その種類等から転送先の負荷の状態を予測し、その予測結果に基づいてデータを転送する。これにより、転送元は、転送先にその負荷の状態や空容量についての問い合わせを行うことなく、負荷の状態や空容量の状態といった転送先の状態に合わせたデータ転送や、更に割り当てられている処理といった自身の状態も考慮したデータ転送が可能となる。

【００１３】

【発明の実施の形態】以下、図面を参照しながら、本発明の実施の形態につき詳細に説明する。

＜第１の実施の形態＞図１は、第１の実施の形態が適用されたシステムの概観図である。

【００１４】図１に示すように、上記システムは、ユーザに演奏操作される鍵盤１０２を備えた電子楽器１０１と、それにケーブル１０４によって接続されたテレビ１０３とから構成されている。第１の実施の形態によるデータ処理装置は、電子楽器１０１に搭載されることでシステムに適用されている。

【００１５】図２は、図１に示すシステムの構成図である。次に、図２を参照して、その構成、及び動作について説明する。第１ＣＰＵ２０１は、ＲＯＭ２０２に格納されているプログラムを読み出して実行することにより、ＲＡＭ２０３をワーク領域として使用しながら楽器１０１全体を制御する。その制御を通して、テレビ１０３にユーザが所望する楽譜等の画像を表示させる。

【００１６】上記ＲＯＭ２０２には、第１ＣＰＵ２０１が実行するプログラムの他に、自動演奏用の演奏データや各種制御用データ、更にはテレビ１０３に画像を表示させるための画像データが格納されている。その画像データは、例えば画像を構成する部分画像（スプライト）毎に用意されている。

【００１７】第１ＣＰＵ２０１は、画像データを表示させる場合、ＲＯＭ２０２から画像データを読み出してそれをＲＡＭ２０３に格納した後、ＶＤＰ（ビデオ・ディスプレイ・プロセッサ）２０４に転送する。また、ＶＤＰ２０４に、各画像データを表示させる座標データも合

わせて転送する。

【００１８】他方のＶＤＰ２０４は、第１ＣＰＵ２０１から転送された画像データをＶ－ＲＡＭ（ビデオＲＡＭ）２０５に格納し、座標データを内部のメモリに格納する。そのＶ－ＲＡＭ２０５に格納した画像データは、座標データに従って読み出し、読み出した画像データを予め定められた優先順位に従って合成（重ね合わせ）する。その合成が行われた画像データは、ＲＧＢデータとしてＶＤＰ２０４から出力される。そのＲＧＢデータは、不図示のエンコーダによってＮＴＳＣ方式の映像信号に変換された後、端子、及び図１に示すケーブル１０４を介してテレビ１０３に出力される。それにより、テレビ１０３の画面に例えば図１に示すような楽譜が表示される。

【００１９】第１ＣＰＵ２０１は、鍵盤１０２に対してユーザが行った操作を、ゲートアレイ（ＧＡ）２０６、及びキーマトリクス回路２０７を介して検出する。より具体的には、キーマトリクス回路２０７を用いて鍵盤１０２をスキャンし、ゲートアレイ２０６を介してスキャン結果を受け取り、その受け取ったスキャン結果を解析することにより、鍵盤１０２に対してユーザが行った操作を検出する。操作を検出した後は、その操作内容に従って楽音を発音させるための制御データ（発音制御データ）を生成し、それを第２ＣＰＵ２０９に転送する。その制御データは、例えばＭＩＤＩ規格に合わせて生成する。

【００２０】スイッチ群２０８は、モードや音色、テンポといった各種の設定用のスイッチをまとめて表したものである。第１ＣＰＵ２０１は、鍵盤１０２に対して行われた操作を検出するのと同様に、キーマトリクス回路２０７を用いてスイッチ群２０８をスキャンし、ゲートアレイ２０６を介してスキャン結果を受け取り、その受け取ったスキャン結果を解析することにより、スイッチ群２０８を構成する各スイッチの操作状態を検出する。操作状態を検出した後は、その操作状態に従って楽器１０１の設定を行う。

【００２１】上記ゲートアレイ２０６は、第１ＣＰＵ２０１と第２ＣＰＵ２０９のインターフェイス等の役割を果たす。第１ＣＰＵ２０１が生成した発音制御データは、ゲートアレイ２０６を介して第２ＣＰＵ２０９に転送される。

【００２２】上記第２ＣＰＵ２０９は、内部に備えたＲＯＭ（図示せず）に格納されているプログラムを実行することにより、第１ＣＰＵ２０１から転送された発音制御データに従って、楽音の発音に関わる制御を実行する。その制御は、ＲＡＭ２１０をワーク領域として使用しながら行う。

【００２３】発音制御データはＭＩＤＩデータの形で第２ＣＰＵ２０９に転送される。第２ＣＰＵ２０９は、第１ＣＰＵ２０１から発音制御データ（ＭＩＤＩデータ）

が転送されると、それを解析して、その発音制御データによって指示される楽音の発音を実現させるための制御データを生成する。生成した制御データはＬＳＩ２１１に送出する。

【００２４】ＬＳＩ２１１は、ＲＡＭ２１１をワーク領域として使用しながら、ＲＯＭ２１３に格納されている波形データを用いて、実際に発音させる楽音の波形データを生成する音源ＬＳＩである。第２ＣＰＵ２０９が送出した制御データに従って、ＲＡＭ２１３から波形データを読み出し、その波形データの加算や、指示された音響効果を付加させるための処理を施した後、Ｄ／Ａコンバータ（ＤＡＣ）２１４に出力する。

【００２５】Ｄ／Ａコンバータ２１４は、ＬＳＩ２１１が出力した波形データをアナログのオーディオ信号に変換してアンプ２１５に出力する。そのアンプ２１５は、Ｄ／Ａコンバータ２１４が出力したオーディオ信号を増幅させてスピーカ２１６に出力する。アンプ２１５からオーディオ信号がスピーカ２１５に入力されることにより、スピーカ２１６から楽音が放音される。

【００２６】上述したように、本実施の形態を搭載した電子楽器１０１は、楽音を発音させるという本来の機能の他に、画像を表示させる機能を備えている。それらの機能を実現させるための負荷は重いことから、第１及び第２ＣＰＵ２０１、２０９の２つのＣＰＵを用意し、各ＣＰＵ２０１、２０９に、その負荷を分散させる構成となっている。即ち第１ＣＰＵ２０１に画像表示を含めた楽器１０１全体の制御を実行させ、第２ＣＰＵ２０９に楽音の発音に関する制御を実行させる構成となっている。

【００２７】次に、図２、図３を参照して、本発明に関わる第１ＣＰＵ２０１が第２ＣＰＵ２０９にデータ転送を行う際の動作について詳細に説明する。上述したように、第１ＣＰＵ２０１は発音制御データを第２ＣＰＵ２０９に転送することにより、第２ＣＰＵ２０９に発音に関する制御を依頼する。第２ＣＰＵ２０９が第１ＣＰＵ２０１から転送された発音制御データの処理に要する時間は、基本的にデータ量とデータの種類によって決まる。このことから、第１の実施の形態では、発音制御データを構成することができるデータの種類毎に、その処理に要する時間をデータ（発音制御データ処理時間データ）として予めＲＯＭ２０２に用意している。

【００２８】図３は、ＲＯＭ２０２に用意された発音制御データ処理時間データ群の構成図である。発音制御データを構成できるデータとしては、例えばＭＩＤＩ規格として定められているように、楽音の発音開始を指示するノートオン、発音中の楽音の消音を指示するノートオフ、音色の切り替えを指示するプログラムチェンジ、主に演奏における音響効果の制御用に用いられるコントロールチェンジ等の数多くの種類がある。それらの種類毎に、第２ＣＰＵ２０９がその処理に要する時間が１６ビ

ットのデータの形でＲＯＭ２０２に用意している。図３に示すように、それらのデータがまとまって発音制御データ処理時間データ群を構成している。その第１ワードデータとしてノートオンの処理時間データ、第２ワードデータとしてノートオフのそれ、第３ワードデータとしてプログラムチェンジのそれが格納されている。

【００２９】なお、第１の実施の形態では、転送した発音制御データを第２ＣＰＵ２０９に確実に処理させるために、各発音制御データ処理時間データを、第２ＣＰＵ２０９がそれに対応するデータの処理に必要な最大時間としている。しかし、最大時間ではなく、例えば平均的な時間を処理時間データとして用意しても良い。

【００３０】図３に示すような発音制御データ処理時間データ群を用意することにより、転送した発音制御データの処理に第２ＣＰＵ２０９が必要としている時間を予測することができる。第１ＣＰＵ２０１は、発音制御データ処理時間データ群を参照し、発音制御データとして転送するデータの種類、及びデータ量に応じて、それに対応する処理時間データを取得して加算や乗算といった演算処理を行うことにより、第２ＣＰＵ２０９が必要とする処理時間を算出する。

【００３１】その処理時間を算出することにより、第２ＣＰＵ２０９に対して空容量（例えば第２ＣＰＵ２０９内部のメモリに割り当てられている領域の空容量である）の問い合わせを行うことなく、それの負荷の状態に合わせて発音制御データを転送することができる。例えば、転送した発音制御データに対する処理が終了するのに合わせて発音制御データを第２ＣＰＵ２０９に転送することもできるようになる。

【００３２】転送された発音制御データの処理が進み、その負荷が小さくなっている状態とは、第２ＣＰＵ２０９が空容量を確保する必要がない状態である。このため、空容量を確保する必要がない状態を選んで発音制御データを転送することにより、問い合わせにかかる時間や、第２ＣＰＵ２０９が空容量を確保するのにかかる時間を確実に省くことができる。それにより、データ転送に要する時間が短縮されるので、各ＣＰＵ２０１、２０９のスループットを向上させることができ、高い処理効率を得ることができる。

【００３３】第１ＣＰＵ２０１が実行する主要なサブルーチン処理としては、スイッチ群２０８を構成している各スイッチの操作状態を検出してそれらの操作状態に応じた設定を行うスイッチ処理、鍵盤１０２に対して行われた操作を検出して発音制御データを生成する鍵盤処理、自動演奏用のデータを再生するシーケンスデータ再生処理等が挙げられる。これらの各処理は、予め定められた順序に従って繰り返し実行されるようになっており、発音制御データはこれらの処理で生成される。なお、本発明の理解を容易にし、また不要な混乱を避けるために、第１ＣＰＵ２０１が実行する主要なサブルーチン処理を、スイッチ処理、鍵盤処理、及びシーケンスデータ再生処理の３つに限定して以降の説明を行うこととする。

【００３４】第１の実施の形態では、上記各処理に必要とする実行時間をデータとしてＲＯＭ２０２に用意している。図４は、ＲＯＭ２０２に用意された主要ルーチン実行時間データ群の構成図である。図４に示すように、各データは各々１６ビットのデータであり、第１ワードデータとしてスイッチ処理に要する実行時間データ、第２ワードデータとして鍵盤処理に要する実行時間データ、第３ワードデータとしてシーケンスデータ再生処理に要する実行時間データが格納されている。その主要ルーチン実行時間データ群を参照することにより、各処理の実行時間を予測することができるようになっている。

【００３５】なお、主要ルーチン実行時間データ群として用意した実行時間データは、全てそれに対応する処理の最大の実行時間としている。しかし、最大の実行時間ではなく、その平均的な実行時間、或いはその最小の実行時間であっても良い。

【００３６】第１の実施の形態では、上記各処理に続けて発音制御データを第２ＣＰＵ２０９に転送する発音制御データ転送処理を実行するようになっている（図５参照）。各データ転送処理では、それに続いて実行される処理の予測される実行時間と、第２ＣＰＵ２０９が転送されている発音制御データに要する予測される残りの処理時間とに基づいてデータ転送を行うようにしている。より具体的には、次に実行する処理で予測される実行時間が、第２ＣＰＵ２０９が必要としていると予測される残りの処理時間以下であった場合にはデータの転送を先送りするようにしている。そのようなデータ転送の先送りを行うことにより、第１及び第２ＣＰＵ２０１、２０９に発生するあそび時間を全体として小さくすることができる。このため、従来と比較して、スループットを向上させることができる。

【００３７】図５は、第１ＣＰＵ２０１が実行する全体処理の動作フローチャートである。次に、図５を参照して、上記第１ＣＰＵ２０１の動作について詳細に説明する。図５に示す全体処理は、第１ＣＰＵ２０１が、ＲＯＭ２０２に格納されているプログラムを読み出して実行することにより実現される処理動作である。

【００３８】楽器１０１の電源がオンされると、始めにステップ５０１のイニシャル処理が実行され、楽器１０１が予め定められた状態に設定される。このとき、制御用の各変数の値も初期値に設定される。その変数として、発音制御データ転送処理以外で実行が終了した直前の処理を示すための変数ｒｏｕｔｉｎｅ＿ｋｉｎｄや、第２ＣＰＵ２０９が発音制御データの処理に必要としていると予測する処理残り時間を示す値が代入される変数ｔｉｍｅ＿ｔｏ＿ｗａｉｔがある。それらの変数には各々０が代入される。

【００３９】ステップ５０１に続くステップ５０２では、スイッチ群２０８をスキャンして各スイッチの操作状態を検出し、それらの操作状態に応じて各種の設定等を行うスイッチ処理を実行する。上記変数ｒｏｕｔｉｎｅ＿ｋｉｎｄには、そのスイッチ処理で０が代入される。スイッチ処理が終了すると、ステップ５０３に移行する。

【００４０】ステップ５０３では、発音制御データを第２ＣＰＵ２０９に転送する発音制御データ転送処理を実行する。詳細は後述するように、変数ｔｉｍｅ＿ｔｏ＿ｗａｉｔに代入された処理残り時間、次に実行される処理で予測される実行時間に基づいてデータ転送が行われる。

【００４１】ステップ５０３に続くステップ５０４では、鍵盤１０２をスキャンしてユーザが行った操作を検出し、その検出した操作に従って発音制御データを生成する鍵盤処理を実行する。この鍵盤処理では、上記変数ｒｏｕｔｉｎｅ＿ｋｉｎｄに１を代入する。その鍵盤処理が終了すると、ステップ５０５で発音制御データ転送処理を実行した後、ステップ５０６に移行する。

【００４２】ステップ５０６では、ユーザがスイッチ群２０８の所定のスイッチを介して行った指示に従って、自動演奏用のシーケンスデータを再生するシーケンスデータ再生処理を実行する。このシーケンスデータ再生処理が実行されることにより、自動演奏による楽音の発音や、テレビ１０３の画面上に自動演奏中の曲の楽譜が表示される。変数ｒｏｕｔｉｎｅ＿ｋｉｎｄには２が代入される。そのシーケンスデータ再生処理が終了すると、ステップ５０７で発音制御データ転送処理を実行した後、ステップ５０２に戻る。

【００４３】このように、第１の実施の形態では、スイッチ処理や鍵盤処理といった主要なサブルーチン処理に続けて発音制御データ転送処理を行っている。各処理の実行時間や第２ＣＰＵ２０９における処理残り時間は予測した時間であり、多少の誤差がありえる。しかし、図５に示すように、主要なルーチン処理と発音制御データ転送処理を交互に行うようにすると、第１及び第２ＣＰＵ２０１、２０９は全体処理を単位としてではなく、その全体処理を構成するルーチン処理単位で処理を行うことになって、発生する遊び時間が全体的に小さくなる。それにより、上記予測誤差が第１及び第２ＣＰＵ２０１、２０９の遊び時間に反映されるのを抑えることができ、スループットをより向上させることができるようになる。

【００４４】次に、上記ステップ５０３、５０５、及び５０７として実行される発音制御データ転送処理について、図６に示すその動作フローチャートを参照して詳細に説明する。

【００４５】先ず、ステップ６０１では、前回に発音制御データ転送処理を行ってから現在までの経過時間、例えば今回の発音制御データ転送処理が図５中のステップ５０３として実行された場合には、ステップ５０７の発音制御データ転送処理を実行してから現在までの経過時間を算出する。その経過時間は、例えばシステムが備えたフリーランニングタイマ、或いは第１ＣＰＵ２０１が備えたタイマ機能を利用して算出する。その算出した経過時間は、変数ｔｉｍｅ＿ｉｎｔｅｒｖａｌに代入されて用いられる。

【００４６】ステップ６０１に続くステップ６０２では、第２ＣＰＵ２０９が必要としていると予測する処理残り時間が経過時間より小さいか否か判定する。それまでに転送した発音制御データを第２ＣＰＵ２０９が全て処理していると予測される場合、その判定はＹＥＳとなってステップ６０３に移行する。そうでない場合には、その判定はＮＯとなってステップ６０４に移行する。

【００４７】ステップ６０３では、第２ＣＰＵ２０９は発音制御データが転送されるのを待っている状態であると予測されることから、処理残り時間を０に設定する。他方のステップ６０４では、処理残り時間から経過時間を引いた値を処理残り時間に設定する。ステップ６０３、或いは６０４が実行されることにより、変数ｔｉｍｅ＿ｔｏ＿ｗａｉｔに新たな値が代入された後、ステップ６０５に移行する。

【００４８】ステップ６０５では、第２ＣＰＵ２０９に転送すべきデータが有るか否か判定する。直前に実行した処理で発音制御データが生成された場合はもとより、それまでに生成した発音制御データのなかで第２ＣＰＵ２０９に転送していないデータが有る場合、その判定はＹＥＳとなってステップ６０６に移行する。そうでない場合には、即ち第２ＣＰＵ２０９に転送すべき発音制御データが無い場合には、その判定はＮＯとなって一連の処理を終了する。

【００４９】ステップ６０６では、上記ステップ６０３、或いは６０４で更新された処理残り時間（変数ｔｉｍｅ＿ｔｏ＿ｗａｉｔの値）が０か否か判定する。発音制御データが転送されるのを第２ＣＰＵ２０９が待っていると予測される場合、その判定はＹＥＳとなってステップ６０９に移行する。そうでない場合には、その判定はＮＯとなってステップ６０７に移行する。

【００５０】ステップ６０７では、変数ｒｏｕｔｉｎｅ＿ｋｉｎｄの値が示す他のサブルーチン処理の実行予測時間を算出する。その算出は、上述したように、ＲＯＭ２０２に格納されている主要ルーチン実行時間データ群を参照し、変数ｒｏｕｔｉｎｅ＿ｋｉｎｄの値に対応する実行時間データを取得することことで行われる。それが終了した後、ステップ６０８に移行する。

【００５１】ステップ６０８では、処理残り時間が、ステップ６０７で取得した実行予測時間より小さいか否か判定する。次のサブルーチン処理の実行が終了する前に第２ＣＰＵ２０９がその処理を完了することが予測され

る場合、その判定はＹＥＳとなってステップ６０９に移行する。そうでない場合には、その判定はＮＯとなって一連の処理を終了する。

【００５２】ステップ６０９では、第２ＣＰＵ２０９に転送すべきデータを実際に転送するデータ転送処理を実行する。そのデータは、第１ＣＰＵ２０１からゲートアレイ２０６を介して第２ＣＰＵ２０９に転送される。そのデータ転送が終了すると、ステップ６１０に移行して、転送したデータに応じた処理残り時間（変数ｔｉｍｅ＿ｔｏ＿ｗａｉｔの値）の更新を行う。ステップ６０１で経過時間を算出するための発音制御データ転送処理の実行時刻は、例えばこのステップ６１０で保持する。そのステップ６１０の処理を実行した後、一連の処理を終了する。

【００５３】このように、発音制御データ転送処理では、第１ＣＰＵ２０１が次に実行する処理と第２ＣＰＵ２０９が実行している処理の何れが早く終了するかにより、データの転送を行うか否かを決定している。このようなデータ転送を行うことにより、第１ＣＰＵ２０１、第２ＣＰＵ２０９の各遊び時間の合計が最も小さくなるように、各ＣＰＵ２０１、２０９に処理を実行させることになる。従って、スループットが向上し、高い処理効率を得ることができる。ユーザは、そのスループットの向上を、楽器１０１に対する操作、例えば鍵盤１０２を操作したときの反応から認識することができる。

【００５４】なお、第１の実施の形態では、発音制御データ転送処理を他のサブルーチン処理と交互に行うようにしているが、必ずしも交互に行うようにしなくとも良い。例えば図５に示す全体処理においては、ステップ５０２のスイッチ処理は他のステップ５０４の鍵盤処理等と比較して生成する発音制御データ量が小さいことから、ステップ５０３の発音制御データ転送処理を行わないようにしても良い。

【００５５】また、主要ルーチン実行時間データ群については、第１の実施の形態では各サブルーチン処理毎に１個の実行時間データだけを用意しているが、設定されているモードに応じて実行時間が大きく異なるサブルーチン処理（例えばシーケンスデータ再生処理）も存在する。このことから、モード別に実行時間データを複数用意し、設定されているモードに応じて実行時間データを参照するようにしても良い。

＜第２の実施の形態＞上記第１の実施の形態では、ＲＯＭ２０２に格納されている主要ルーチン実行時間データ群を使用している。しかし、各サブルーチン処理の実際の実行時間は、ユーザの楽器１０１に対する操作に応じて変化することから、その主要ルーチン実行時間データ群としてＲＯＭ２０２に用意した各実行時間データが必ずしも最適であるとは限らない。各実行時間データをより最適な値とすることにより、各ＣＰＵ２０１、２０９のスループットをより向上させることができる。第２の

実施の形態は、このことに着目して、各サブルーチン処理のより最適な実行時間を予測できるようにしたものである。

【００５６】第２の実施の形態が適用されたシステム構成は、基本的に第１の実施の形態が適用されたそれと同じであり、また、処理動作も基本的に同じである。このため、第１の実施の形態で使用した符号をそのまま使用して、第１の実施の形態から異なる部分のみ説明することにする。

【００５７】第２の実施の形態では、ＲＯＭ２０２に格納された固定値データ群としての主要ルーチン実行データ群（図４参照）に加えて、各サブルーチン処理の実際の実行時間を計測して取得し、それをＲＡＭ２０３に格納して参照するようにしている。図７は、そのようにしてＲＡＭ２０３に格納された主要ルーチン実行時間データ群の構成図である。

【００５８】図７に示すように、各サブルーチン処理毎に格納領域を用意し、計測した実際の実行時間を対応する格納領域内に格納している。第２の実施の形態では、各格納領域に５個の実行時間データを格納できるようにしている。５個の実行時間データを格納した後は、そのなかで最古の時間データと置き換える。これにより、常に最新の５個の実行時間データを保持するように、実行時間データの更新を行っている。

【００５９】第２の実施の形態において、各サブルーチン処理の実際の実行時間は、例えば以下のようにして計測される。図８を参照して説明する。図８は、図５に示す全体処理内でステップ５０２として実行されるスイッチ処理の動作フローチャートである。

【００６０】先ず、ステップ８０１では、現在のタイマ（例えばフリーランニングタイマ）値をセーブする。それに続くステップ８０２では、スイッチ処理におけるメイン処理を実行する。そのステップ８０２は、第１の実施の形態におけるステップ５０２と同じ処理である。そのメイン処理が終了すると、ステップ８０３に移行し、現在のタイマ値からステップ８０１でセーブしたタイマ値を減算して今回の実行時間データを算出する。その算出した実行時間データは、ＲＡＭ２０３のスイッチ処理用の格納領域に格納する。このとき、５個の実行時間データが格納されていなければ空いている場所に、５個格納されていればそのなかで最古の実行時間データが格納されている場所に今回取得した実行時間データを格納する。その実行時間データの格納が終了した後、一連の処理を終了する。

【００６１】このように、スイッチ処理の始めと終わりの各時点でのタイマ値を参照することで、そのスイッチ処理の実行時間を算出することができる。他の鍵盤処理、及びシーケンスデータ再生処理における実行時間の取得方法も同様であるため、その詳細な説明は省略する。

13

【００６２】第１の実施の形態において、次のサブルーチン処理の実行時間の予測は図６に示す発音制御データ転送処理内のステップ６０７で行われる。第２の実施の形態では、ＲＯＭ２０２、及びＲＡＭ２０３にそれぞれ主要ルーチン実行時間データ群を格納していることから、ステップ６０７では以下のようにして実行予測時間を算出する。

【００６３】楽器１０１の電源がオンされた時点では、ＲＡＭ２０３には主要ルーチン実行時間データ群として格納されたデータが存在しない。このため、ＲＯＭ２０２に格納されている主要ルーチン実行時間データ群を参照して実行予測時間を算出する。第２の実施の形態では、５個の実行時間データを取得していないサブルーチン処理の実行予測時間はＲＯＭ２０２に格納された実行時間データを用いて算出する。

【００６４】５個の実行時間データを取得したサブルーチン処理に対しては、ＲＯＭ２０２に格納されている実行時間データを参照せずに、その５個の実行時間データに基づいて実行予測時間を算出する。具体的には、５個の実行時間データのなかで最大の実行時間データを抽出し、その実行時間データを用いて実行予測時間を算出する。

【００６５】このようにして実行予測時間を算出することにより、サブルーチン処理の実際の実行時間が反映させた発音制御データの転送を行うことができる。そのデータ転送は、過去の履歴に基づいていることから、現実に即して行うことができる。ステップ６０８の判定処理による分岐がより高精度に行われるようになる。このため、第１の実施の形態と比較して、各ＣＰＵ２０１、２０９のスループットをより向上させることができる。

【００６６】なお、第２の実施の形態では、実行時間データとして取得する個数を５個としているが、それ以外の数としても良い。それらの実行時間データのなかで使用するものは、第２の実施の形態のように最大値ではなく、それらの平均値としても良い。

【００６７】また、実行時間は、サブルーチン処理を実行する度に取得するようにしているが、間欠的に実行時間データを取得するようにしても良い。設定されているモードの違いに対応できるように、取得した実行時間をモード別に管理（保持）するようにしても良い。

＜第３の実施の形態＞上記第１及び第２の実施の形態は、共に、サブルーチン処理の実行予測時間と処理残り時間とを比較して、その比較結果から第２ＣＰＵ２０９にデータ転送を行うか否かを判定している。しかし、その判定は、各ＣＰＵ２０１、２０９の遊び時間を共に小さくすることに着目したものであり、第２ＣＰＵ２０９が転送済みの発音制御データの処理に要する負荷の重さが必ずしも反映されていない。第３の実施の形態は、その第２ＣＰＵ２０９の現在の負荷の重さを更に考慮してデータ転送を行うようにしたものである。

14

【００６８】第３の実施の形態が適用されたシステム構成は、基本的に第１、或いは第２の実施の形態が適用されたそれと同じであり、また、処理動作も基本的に同じである。このため、第１の実施の形態で使用した符号をそのまま使用して、第１、或いは第２の実施の形態から異なる部分のみ説明することにする。

【００６９】第３の実施の形態では、図６に示す発音制御データ転送処理内のステップ６０９のデータ転送処理だけが第１、或いは第２の実施の形態から異なっている。このため、そのステップ６０９のデータ転送処理についてのみ、図９に示すその動作フローチャートを参照して詳細に説明する。

【００７０】なお、ステップ６０９のデータ転送処理は、第２ＣＰＵ２０９で予測される処理残り時間が０ではなく、且つ第２ＣＰＵ２０９に転送すべき発音制御データがある場合に実行される処理である。処理残り時間は０でないことから、その値はステップ６０４で更新された値である。

【００７１】先ず、ステップ９０１では、ステップ６０４で更新された処理残り時間が、所定時間より大きいか否か判定する。その所定時間は、第２ＣＰＵ２０９の現在の負荷が重い状態か否かを判定するための予め定めた定数である。このため、第２ＣＰＵ２０９が現在重い状態であると予測される場合、その判定はＹＥＳとなってステップ９０２に移行する。そうでない場合には、その判定はＮＯとなってステップ９０３に移行する。

【００７２】ステップ９０２では、第２ＣＰＵ２０９の現在の負荷が重い状態となっていることから、第２ＣＰＵ２０９に転送すべき発音制御データのなかから、ノートオンデータを削除する。その後、ステップ９０２に移行する。

【００７３】なお、ステップ９０２でノートオンデータを削除するのは、主に、ノートオンデータの処理に必要な処理時間が比較的に大きいこと、仮にそれを転送しなくとも発音全体に重大な影響を与えない（ここでは、発音がホールドする（鳴りっぱなしになる）といったことがないといった意味である）、という２つの理由からである。ノートオンデータの削除は、例えばＲＡＭ２０３内に確保した第２ＣＰＵ２０９に転送すべき発音制御データを格納する領域（バッファ）から、ノートオンデータを削除することで行われる。

【００７４】ステップ９０３は、第２ＣＰＵ２０９に転送すべき発音制御データが格納した後に行われる処理である。そのステップ９０３では、その発音制御データを第２ＣＰＵ２０９に実際に転送する処理を実行する。その発音制御データを第２ＣＰＵ２０９に転送する処理が、このデータ転送処理におけるメイン処理である。そのデータの転送が終了した後、一連の処理を終了する。

【００７５】このように、第３の実施の形態では、第２ＣＰＵ２０９の負荷の状態に応じて、転送すべきデータ

を選別して第２ＣＰＵ２０９に転送する。このため、第２ＣＰＵ２０９の負荷が非常に重くなる状況を回避することができ、結果として、各ＣＰＵ２０１、２０９がより効率的に処理を行うことができるようになる。

【００７６】なお、第３の実施の形態では、第２ＣＰＵ２０９の負荷が重い状態と予測される場合、ノートオンデータを削除するようにしているが、そのデータを削除するのではなく、転送を先送りするようにしても良い。また、削除、或いは先送りするデータとしては、ノートオンデータではなく、発音全体に対する影響が比較的に小さい他の種類のデータとしても良い。データを生成した順序に従って、第２ＣＰＵ２０９の負荷の状態に応じて随時転送するようにしても良い。削除、或いは先送りするデータの種類は、第２ＣＰＵ２０９の負荷の状態に応じて決めるようにしても良い。

【００７７】また、本実施の形態（第１～第３の実施の形態）においては、第２ＣＰＵ２０９の負荷の状態を予測だけしているが、その予測誤差を小さくするために、例えばある時間間隔毎といったように、第１ＣＰＵ２０１が第２ＣＰＵ２０９に対して間欠的にその負荷の状態（未処理のデータ量等）を問い合わせて、予測誤差を修正するようにしても良い。これにより、第２ＣＰＵ２０９の状態を常時より高精度に予測することができるようになる。

【００７８】本実施の形態のＣＰＵ数は２個であるが、当然のことながら、本発明が前提としているＣＰＵ数はこれに限定されるものではない。また、本発明は、新製品だけでなく、所定の記憶媒体、或いは通信手段を介して本発明の機能を実現できるプログラムを配布することにより、既存の装置であっても本発明を実現させることができる。電子楽器以外でも本発明を適用することは容易である。

【００７９】

【発明の効果】以上説明したように本発明では、転送元（第１ＣＰＵ）がデータの転送先（第２ＣＰＵ）にデータを転送する場合、転送先の処理能力、転送済みのデータ量、その種類等から転送先の負荷の状態を予測し、その予測結果に基づいてデータを転送する。このため、転送元は、転送先にその負荷の状態や空容量についての問い合わせを行うことなく、負荷の状態や空容量の状態といった転送先の状態に合わせたデータ転送や、更に割り当てられている処理といった転送元自身の状態も考慮したデータ転送を行うことができる。

【００８０】それらのようなデータ転送では、問い合わせに要する時間や、転送先が空容量の確保に要する時間を省くことができる。そのため、データ転送にかかる時間を短縮することができる。そのデータ転送にかかる時間が短縮される結果、転送元、及び転送先のスループットも向上させることができる。

【図面の簡単な説明】

【図１】本実施の形態が適用されたシステムの概観図である。

【図２】本実施の形態が適用されたシステムの構成図である。

【図３】発音制御データ処理時間データ群の構成図である。

【図４】主要ルーチン実行時間データ群の構成図である。

【図５】第１ＣＰＵが実行する全体処理の動作フローチャートである。

【図６】発音制御データ転送処理の動作フローチャートである。

【図７】主要ルーチン実行時間データ群の構成図である（第２の実施の形態）。

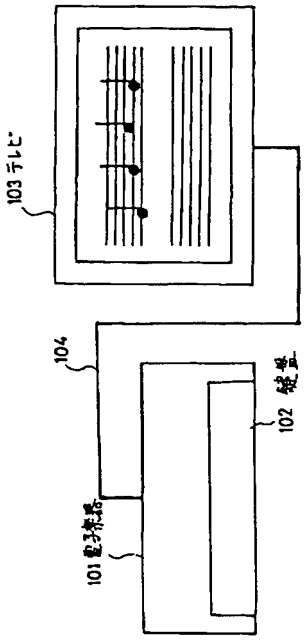
【図８】スイッチ処理の動作フローチャートである（第２の実施の形態）。

【図９】データ転送処理の動作フローチャートである（第３の実施の形態）。

【符号の説明】

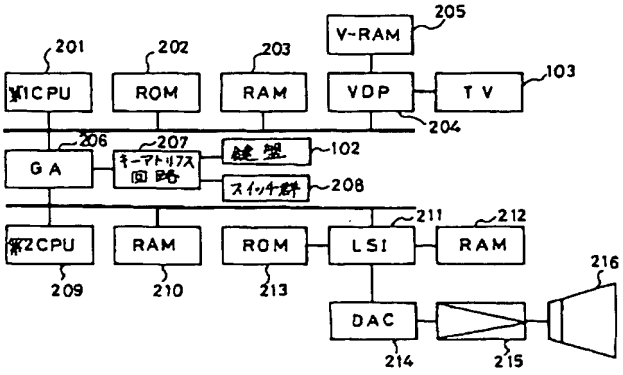
１０１　電子楽器
１０２　鍵盤
１０３　テレビ
２０１　第１ＣＰＵ
２０２　ＲＯＭ
２０４　ＶＤＰ
２０８　第２ＣＰＵ
２１０　ＬＳＩ

【図１】



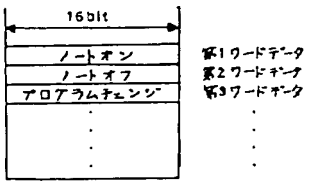
第1の実施の形態が適用されたシステムの概観図

【図２】



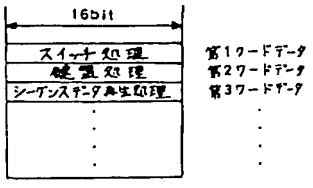
第1の実施の形態が適用されたシステムの構成図

【図３】



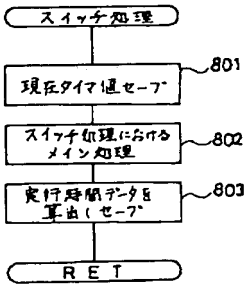
発音制御データ処理時間データ群の構成図

【図４】



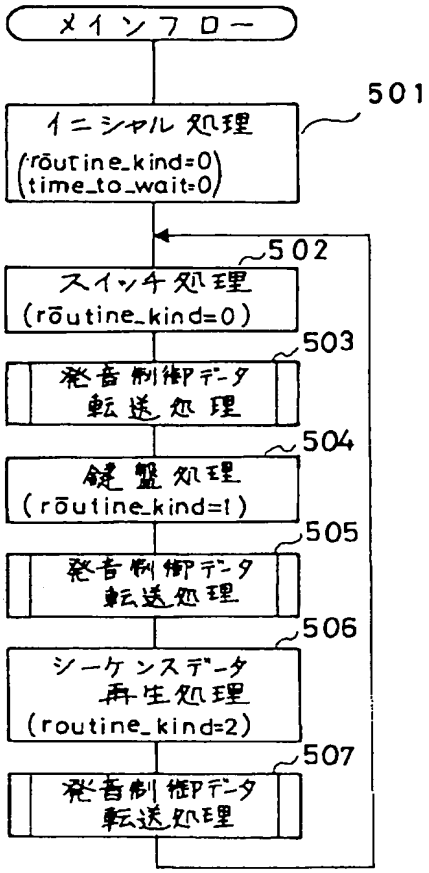
主要ルーチン実行時間データ群の構成図

【図８】



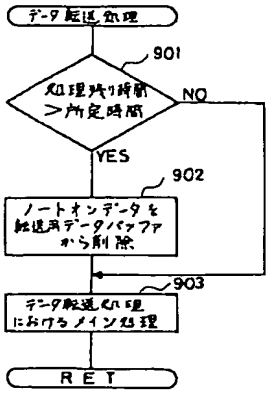
スイッチ処理の動作フローチャート(第2の実施の形態)

【図５】



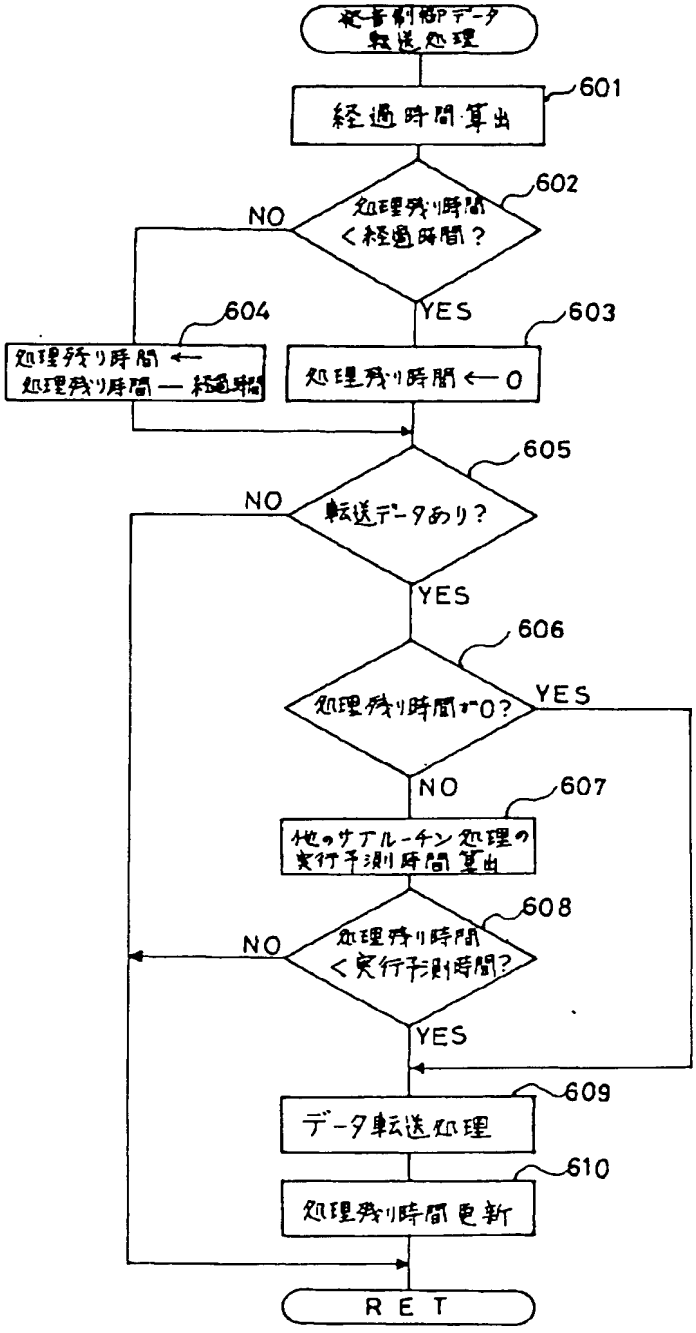
第１ＣＰＵが実行する全体処理の動作フローチャート

【図９】



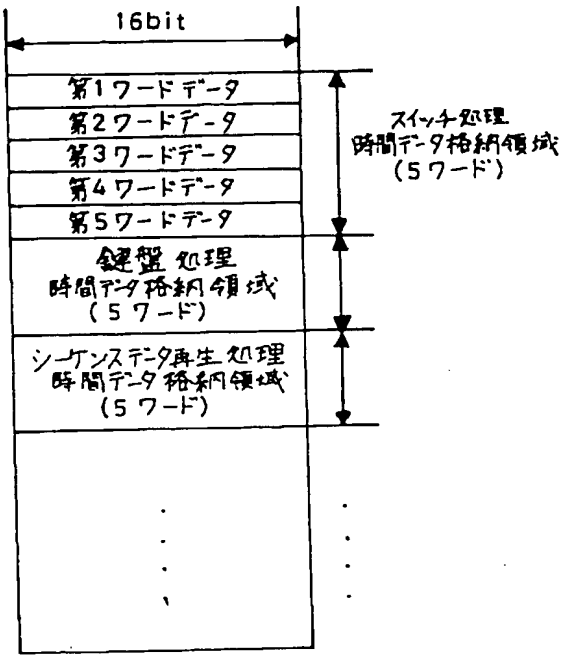
データ転送処理の動作フローチャート(第３の実施の形態)

【図６】



発音制御データ転送処理の動作フローチャート

【図７】



RAMに格納された主要ルーチン実行時間データ群の構成図（第２の実施の形態）